

Enter Web Address:     Searched for <http://www.ibm.com/developerworks/web/library/wa-resc/>

13 Results

\* denotes when site was updated.

Material typically becomes available here 6 months after collection. [See FAQ](#).**Search Results for Jan 01, 1996 - Oct 02, 2008**

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
0	0	0	0	0	4 pages	1 pages	1 pages	3 pages	1 pages	1 pages	2 pages	0 pages

May 26, 2001 *	<a href="#">Feb 13, 2002</a>	<a href="#">Jun 08, 2003 *</a>	<a href="#">Jan 04, 2004</a>	<a href="#">Mar 16, 2005 *</a>	<a href="#">Apr 28, 2006 *</a>	<a href="#">Oct 13, 2007 *</a>
<a href="#">Jun 17, 2001</a>			<a href="#">May 10, 2004</a>			<a href="#">Oct 13, 2007 *</a>
<a href="#">Aug 21, 2001 *</a>			<a href="#">Dec 15, 2004</a>			
<a href="#">Nov 19, 2001</a>						

[Home](#) | [Help](#)[Internet Archive](#) | [Terms of Use](#) | [Privacy Policy](#)

Years ago, Erik wrote some articles about using Remote Scripting in web browsers. These articles are available at the following links:

- [Remote scripting using a servlet](#)
- [Sending rich messages between client and server using asynchronous messaging](#)
- [Pushing Messages to a Browser Using Remote Scripting](#)

The online demo of remote scripting is no longer hosted here. Please refer to the articles above for the code and documentation. Also, please do not contact Erik directly with support questions regarding remote scripting.

Enter Web Address:  All Take Me Back Adv. Search  Compare Archive PagesSearched for <http://www.chatchersolutions.com/rs/>

19 Results

\* denotes when site was updated.

Material typically becomes available here 6 months after collection. [See FAQ](#).**Search Results for Jan 01, 1996 - Oct 02, 2008**

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
0	0	0	0	0	3 pages	3 pages	3 pages	5 pages	1 pages	0	4 pages	0
pages	pages	pages	pages	pages						pages		pages

<a href="#">Jun 09, 2001</a>	<a href="#">Feb 13, 2002</a>	<a href="#">May 14, 2003</a>	<a href="#">Jan 03, 2004</a>	<a href="#">Mar 16, 2005</a>	<a href="#">Feb 22, 2007</a>
<a href="#">Nov 04, 2001</a>	<a href="#">Jun 03, 2002</a>	<a href="#">Jun 08, 2003</a>	<a href="#">Mar 02, 2004</a>	<a href="#">May 24, 2007</a>	
<a href="#">Dec 27, 2001</a>	<a href="#">Oct 09, 2002</a>	<a href="#">Oct 15, 2003</a>	<a href="#">Jun 04, 2004</a>	<a href="#">May 25, 2007</a>	
			<a href="#">Nov 23, 2004</a>	<a href="#">Oct 21, 2007</a>	
			<a href="#">Dec 11, 2004</a>		

[Home](#) | [Help](#)[Internet Archive](#) | [Terms of Use](#) | [Privacy Policy](#)

# Remote Scripting

Erik's Articles about Remote Scripting

- [Remote scripting using a servlet](#)
- Two more articles on the way! Stay tuned...

Examples from the articles

- Example from the first article coming soon
- [RSMessaging](#) (from the second up and coming developerWorks article).
- [Javadocs](#)

Resources

- Read Microsoft's documentation on [remote scripting](#).
- Download [Brent Ashley's JSRS](#).



IBM developerWorks

[Search help](#) |  
[Feedback](#)

## developerWorks **search results**

Search for:  within 

32 results for Hatcher within All of dW

[\\* Show me dW forum search results](#)[New site feature](#)**Firefox users:**[Click to add the dW search plug-in to your Firefox browser](#)

Sort by relevance

Sort by date

**1. [Remote scripting using a servlet](#)**

... in typical standalone or client-server desktop applications because of the constraints that HTML and HTTP impose. Here, Erik **Hatcher** explains how remote scripting can be used to enhance the interactivity and dynamic nature of a Web application experience. ... because of the constraints that HTML and HTTP impose. Here, Erik **Hatcher** explains how remote scripting can be used to enhance the interactivity ... for the Tucson Developer Series. You can reach him at [erik@hatcher.net](mailto:erik@hatcher.net). Please take a moment to complete this form to help us ...

01 Feb 2001

**2. [Sending rich messages between client and server using asynchronous messaging](#)**

[article](#) [wa-rich-rsmg.zip](#) [Description](#)[Name](#)[Size](#)[Download](#) method Erik **Hatcher** has been dot.bombed twice this year, and each time he has written ... the revamping of their XML exam. He can be reached at [erik@hatcher.net](mailto:erik@hatcher.net). Please take a moment to complete this form to help us ...

01 May 2001

**3. [Automating the build and test process](#)**

... and throw in automatically generated e-mail reports -- and you'll be well on your way to XP nirvana. Follow along as Erik **Hatcher** shows you how he has modified the popular Ant 1.3 and the JUnit test framework for complete, customized automation of the build ... way to XP nirvana. Follow along as Erik **Hatcher** shows you how he has modified the popular ... JUnit Test Results: `$ {TODAY}" from="erik@hatcher.net"><fileset dir="."> <include name ... j-UnitMail.zip` [Name](#)[Size](#)[Download](#) method Erik **Hatcher** has had a wild ride on the dot-com roller ...

14 Aug 2001

**4. [Implementing Internet call management](#)**

The might of the Internet has been applied to many different business areas, from finance to HR. Here, Jon **Hatcher** introduces the benefits and implementation issues of an Internet call centre, focusing on the server-side technologies that glue ... business areas, from finance to HR. Here, Jon **Hatcher** introduces the benefits and implementation ... PerlHandler Apache::SendSessionKey::handler Jon **Hatcher** has been working on an intranet call management ... his 2nd year. You can reach Jon at [jon.hatcher@uk.ibm.com](mailto:jon.hatcher@uk.ibm.com). Please take a moment to complete ...

01 Dec 2001

**5. [Test flexibly with AspectJ and mock objects](#)**

Programmers who have incorporated unit testing into their development process know the advantages it brings: cleaner code, courage to refactor, and higher velocity. But even the most die-hard unit testers can falter when faced with testing a class that ...

01 May 2002

**6. [An excerpt from "Java Tools for Extreme Programming"](#)**

March 2001) provides an overview of XP and explains why it is important. "Automating the build and test process" by Erik **Hatcher** (developerWorks , August 2001) offers a hands-on look at using Ant and JUnit to automate a process that captures pertinent ...

01 Jul 2002

**7. [Diagnosing Java code: Assertions and temporal logic in Java programming](#)**

developerWorks , June 2001). Follow along in "Automating the build and test process" (developerWorks , August 2001) as Erik **Hatcher** shows you how he has modified the popular Ant 1.3 and the JUnit test framework for complete, customized automation of the build ...

01 Jul 2002

**8. [Diagnosing Java code: Unit tests and automated code analysis working together](#)**

Unit testing and static analysis are often seen as unrelated ways to help ensure the correctness of a program. This article examines the relationship between these two methods and covers how the tools

01 Oct 2002



# JavaScript EE, Part 1: Run JavaScript files on the server side

Learn how to use the javax.script API in Ajax and Java EE applications

Level: Intermediate

Andrei Cioroianu, Senior Java Developer and Consultant, Devsphere

16 Dec 2008

Combine JavaScript with Java™ code on the server to get the freedom to use the same JavaScript routines on both servers and clients. In addition, the techniques presented throughout this series will allow you to maintain a single code base for both Ajax and non-Ajax clients. Because much of the server-side code would still be written in the Java language, you'll find it necessary to expose the Java Platform, Enterprise Edition (Java EE) features to JavaScript. In this series, learn how to run JavaScript files on the server side, call remote JavaScript functions with Ajax, and use the Java Scripting API with the JavaServer Pages (JSP) technology.

Typical Ajax applications use JavaScript on the client side and a different language, such as Java, on the server side. As a result, developers must implement some of their routines twice, using JavaScript for the Web browser and another language for the server. This double-coding issue can be avoided by using JavaScript combined with Java code on the server side, getting full support of scripting languages through the `javax.script` API. In addition, the Java SE Development Kit (JDK) 6 already includes Mozilla's Rhino JavaScript engine, which means no setup is required.

In this first article of the series, you will use a simple script runner that lets you execute JavaScript files within a Java EE application. Scripts will have access to the so-called "implicit objects" that are used in JSP pages, such as `application`, `session`, `request`, and `response`. Most of the samples consist of reusable code so that you can easily start using JavaScript on the server in your own applications.

## Using the `javax.script` API

This section provides an overview of the `javax.script` API. You will learn how to execute scripts that access Java objects, invoke JavaScript functions from your Java code, and implement a caching mechanism for the compiled scripts.

### Executing scripts

The `javax.script` API is very simple. You start by creating a `ScriptEngineManager` instance, which lets you obtain a `ScriptEngine` object (see Listing 1), using one of the following methods:

- `getEngineByName()`
- `getEngineByExtension()`
- `getEngineByMimeType()`

#### Listing 1. Getting a `ScriptEngine` instance

```
import javax.script.*;  
...  
ScriptEngineManager manager = new ScriptEngineManager();  
ScriptEngine engine = manager.getEngineByName("JavaScript");  
...  
engine.eval(...);
```

You can also obtain the list of available script engines with `getEngineFactories()`. Currently, only the JavaScript engine is bundled with JDK 6, but `ScriptEngineManager` implements a discovery mechanism for third-party engines that support *JSR-223 Scripting for the Java Platform* (see [Resources](#)). You just have to place the JAR files of the script engines in your CLASSPATH.

After you get the `javax.script.ScriptEngine` instance, you can call `eval()` to execute scripts. You can also export